REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
		Final 01 Mar 95 - 28 Feb 98

4. TITLE AND SUBTITLE
Combinatorial Optimization with Applications to Resource Management in Communications Networks

5. FUNDING NUMBERS
DAAH04-95-1-0121

6. AUTHOR(S)
Serge Plotkin

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Stanford University

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 277090-2211

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
ARO 32360.1 - MA

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The main goal of this project is to develop fundamental algorithmic techniques that can be applied to problems that arise in the context of high speed communications networks. The emphasis is on efficient algorithms for resourse management.

This research project has two main components. The first is development of well founded algorithmic techniques for online resource allocation where one needs to make decisions based on partial data and without knowledge of the future. The main goal is to develop techniques that lead to provable guarantees on worst case performance and ensures good performance on average. Research on these problems consists of both theoretical analysis and simulation based studies.

The second component is design of efficient offline resourse allocation algorithms based on multicommodity flow techniques. The difference between this work and existing efforts is that we are designing and tuning the algorithms to produce approximate solutions instead of designing algorithms to produce exact solutions in theory. The result is code that is orders of magnitude faster than existing code and which achieves a precision of better than 1%.

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UL	

Final Report

ARO NUMBER:	32360-MA
PERIOD COVERED BY REPORT:	February 95 – February 98
GRANT NUMBER:	DAAH04-95-1-0121
NAME OF THE INSTITUTION:	Stanford University
PROJECT TITLE:	Combinatorial Optimization with Applications to Resource Management in Communication Networks
PI:	Serge Plotkin Dept. of Computer Science, Stanford University (415)-723-0540 plotkin@cs.stanford.edu
STUDENTS SUPPORTED:	Ashish Goel, Omri Palmon, Jeffrey Oldham, Adam Meyerson, Tim Roughgarten, Rene Schaub.
PHD DEGREES GRANTED:	3
MSC DEGREES GRANTED:	2

Hand delivered by Mary Jackson

1 Research Objectives and Motivation

The main goal of the project is to develop fundamental algorithmic techniques that can be applied to problems that arise in the context of high-speed communication networks. The explosion in the size and complexity of networks, together with QoS requirements needed by some of the new services raises many new challenging problems. In particular, as complexity of the networks increases, efficient resource management becomes more important.

Our research project has two main components. The first component is development of well-founded algorithmic techniques for *online* resource allocation, where one needs to make decisions based on partial data and without knowledge of the future. In particular, we focus on online routing and admission control. The main goal here is to develop techniques that lead to provable guarantees on worst-case performance while at the same time ensuring good performance on the average. We consider bandwidth allocation problems in the contexts of virtual circuit routing, large file transfers, and multicast. Our research on these problems consists of both theoretical analysis and simulation-based studies.

The second component of our research effort is design of efficient *offline* resource allocation algorithms, based on multicommodity flow techniques. The main difference between our algorithms and currently existing ones is that, from the beginning, we are designing and tuning the algorithms to produce approximate solutions, instead of designing algorithms that should produce exact solutions in theory. This allows us to produce code that is orders of magnitude faster than previously existing code for this type of problems, while achieving precision of better than 1%.

2 Technical Approach

2.1 Offline Optimization

Our main effort in the context of offline optimization was directed towards design, analysis, and implementation of approximation algorithms for multicommodity flow and related problems. Multicommodity flow is an extension of the classical single-commodity network flow problem, where instead of a single source-sink pair we are given a set of pairs with a demand associated with each pair. The goal is to satisfy these demands by sending flow along the edges of the graph while maintaining capacity constraints, i.e. maintaining that the sum of the flows on any edge is below the capacity of this edge. Multicommodity flow is a fundamental optimization problem with many application. Variants of this problem are often encountered in the context of design and optimization of high-speed networks. Examples of problems that can be viewed as variants of multicommodity flow include routing, admission control, and design of networks with given connectivity parameters.

Since multicommodity flow can be expressed as a Linear Program, classical approaches to solving this problem are based on variants of LP algorithms, such as Simplex or Interior Point algorithms. These algorithms are relatively slow, because of the large number of variables needed to express the problem. Our approach is to design fast *approximation* algorithms. Roughly speaking, although our algorithms can not compute exact solution in polynomial time, they (provably) achieve good approximations very quickly.

Our algorithms are based on starting with some (bad) flow, assigning weights to edges that are *exponential* in the congestion on these edges, and rerouting commodities to less heavy paths. Note that since each rerouting changes congestion on some of the edges, the weights are recomputed after each rerouting, changing the relative weight of different paths.

We have studied the theoretical aspect of these algorithms in [KPST94, LMP+95, PST95, KP95]. These papers show that by carefully choosing the rerouting strategy one can obtain algorithms with worse case behavior that is much better than the worse case behavior of the interior-point based algorithms, such as the one in [KV86, Vai87]. Recently, we have been working on implementing a variant of a min-cost multicommodity flow algorithm based on rerouting according to an exponential function. Our experience shows that implementing the algorithms described in these papers directly does not lead to acceptable performance, as compared to the other existing (and implemented) algorithms for this problem. Our goal was to examine the reasons for the diminished performance and modify the algorithms so that they will work better on average data while maintaining provable worst case performance.

Our implementation, described in [GOPS98], uses a combination of ideas from our earlier theoretical papers [PST95, KP95]. In order to increase performance, we had to redesign these algorithms and make them more "opportunistic". One main change involves the choice of the coefficient in the exponent. The theory predicts that using high enough coefficient ensures that each rerouting results in progress. Unfortunately, this progress is inversely proportional to this coefficient. We have devised an adaptive procedure that adjusts this coefficient so that it will be as large as possible, while ensuring progress. Other improvements included heuristics for choosing which commodities to reroute at each step (instead of the round-robin approach used in the theoretical algorithms) and heuristics for choosing the amount that is rerouted at each step.

In order to assess the performance of our algorithm, we tested it on problems created by generators RMFGEN and by MULTIGRID. We compared the performance of our algorithm to the PPRN algorithm described in [CN96]. According to [CN96], the PPRN implementation that we are using is faster than MINOS 5.3 (general-purpose linear programming package), MCNF85 (special purpose min-cost multicommodity flow program), and LoQo (primal-dual interior point implementation).

Following are representative running times, obtained on SUN Ultra-2, 200MHz, 256Mbytes RAM. Here, n is the number of nodes, m is the number of edges, and k is the number of commodities.

n/m/k	PPRN	MCMCF (1%)	speedup
65/328/100	72.4	1.7	43
145/488/100	475.2	2.4	195
257/712/100	2644.9	3.75	705
577/1352/100	25366.3	13.6	1870
1025/3072/20	881.4	7.9	110

As it can be seen from the above table, our algorithm runs several orders of magnitude faster, while producing solutions that are within 1% of the optimum. It is important to note that the numbers in the above table do not include problems for which we were not able to obtain a solution from PPRN in a day or two of execution. For example, for one of the problems with 1025 nodes, 2248 edges and 100 commodities, our algorithm produced a solution that is within 1% of optimum in about 32 seconds, while PPRN did not produce any solution after 18 hours.

We are currently working on further improving the performance and modifying the algorithm to solve more general packing problems.

2.2 Online strategies

In the context of online computation we focused on developing algorithms for online resource management in communication networks. Together with Tardos, we considered the problem of scheduling and routing FTP transfers in a network that supports per-flow routing.

More precisely, we considered the following problem:

- Request arrive online, each request specifies source and destination nodes and the size of the file to be transmitted.
- After receiving a request we need to decide on:
  - Route that will be used to transmit the file.
  - Rate that the file should be transmitted at.
- At any point, the sum of the transmission rates assigned to any link should not exceed the capacity of this link.
- The overall goal is to minimize average "transmission completed" times.

The idea is to model FTP or WEB data transfers, where the main performance measure is the completion time, eg. how much time it takes to download a series of WEB pages.

Although the above problem is related to the problems studied in [AAP93, AAF+97], the resource allocation strategies developed there can not be applied directly. The main difference lies in the fact that here the data rate of the sources is not given but has to be decided on a per-source basis by the online algorithm. It is easy to see that a greedy approach, where each transfer is assigned maximum currently possible data rate, can not result in good overall performance.

We showed that it is possible to achieve $O(\log^2 n)$ competitive algorithm for this problem, where n is the number of nodes in the network. In other words, the average completion time achieved by our algorithm is at most $O(\log^2 n)$ times larger than the average completion time that can be achieved by the optimum offline algorithm that has a complete knowledge of future arrivals.

2.3 Randomized Online Arrivals

The theoretical competitive ratio of our online FTP routing algorithm mentioned above is too large to make this approach practical. In fact, the theoretical competitive ratios of all the online routing algorithms [AAP93, AAF+97] are not practical. For example, consider a very small network with 4 nodes and durations from 1 to 128 units. If a request sequence can be completely satisfied by the adversary, the algorithm in [AAP93] guarantees to accept at least 3% of the circuits, which is not very useful from the practical point of view. Unfortunately, the lower bounds in [AAF+97, AAP93] imply that one can not do better in this model. In fact, if the holding time of a circuit is unknown at the time the request to route this circuit is issued, it is possible to prove even stronger lower bounds [ABK92, MP97].

On the other hand, we observed in simulations [GKPR94] that our exponential-weight framework leads to throughput that is within a very small constant factor from the *fractional optimum*, i.e. from the bound that is computed by looking at data as flow of some liquid, disregarding "one request - one route" constraints.

In [KPP96] we have explained the apparent contradiction between the competitive bounds and the simulation results for throughput-maximization online routing. We showed that if the requests for connections between any pair of points are generated by a Poisson process and if the holding time of connections is distributed exponentially, then the performance of our online algorithm is much better than $O(\log n)$. Let $\epsilon = \sqrt{r \log n}$, where r the maximum fraction of an edge bandwidth that can be requested by a single circuit. We showed that if an offline algorithm can achieve expected rejection ratio R^*, then our algorithm achieves expected rejection ratio of at most $R^* + O(\epsilon)$.

In contrast to the previously proposed strategies in the Poisson-arrivals model [Kel86, KO88, OK85], we do not assume that the processes describing utilization of different edges are independent. Moreover, contrary to these papers, we assume that the traffic matrix (rates of arrivals between source/destination pairs) is *unknown* and chosen by the adversary.

2.4 Min-Cut / Max-Flow Relationships for Multicommodity Flow

In order to prove that a given multicommodity flow problem is infeasible, it is sufficient to exhibit a cut whose capacity is below the sum of the demands that are separated by the cut. The min-cut max-flow theorem for the single-commodity flow problem states that the non-existence of such a "bad" cut proves that a feasible flow does exist. This theorem, discovered in the fifties, is the basis of currently fastest single-commodity flow algorithms.

For multicommodity flow the situation is more complicated. A multicommodity flow problem can be infeasible even if the "cut condition" is satisfied. A natural question to ask is how large a "safety margin" do we need, *i.e.* how large should be the minimum ratio (over all cuts) of the capacity of the cut to the sum of the demands that are separated by this cut, in order to ensure existence of a feasible flow. A related problem is to consider a multicommodity flow problem, and to search either for a feasible flow, or for a cut whose ratio is below the above mentioned safety margin. This leads to an algorithm that finds an *approximately minimum-cut*. Approximately minimum-cut computation is a basic step for construction of approximation algorithms for a variety of NP-complete problems.

The best bound on the minimum-cut maximum-flow ratio (and the approximation ratio of the associated algorithm) for general multicommodity flow problems was $O(\log n \log D)$, where D is the sum of all the demands and n is the number of nodes in the graph. Note that $\log D$ can be as large as n, making this bound useless. In [PT95] we have improved this bound to $O(\log^2 k)$, where k is the number of commodities, proving for the first time that the bound does not depend on the precision of the input data. We also we proved that if the network is sufficiently sparse, then the bound changes to $O(\log k)$. For planar graphs when the demands are uniform (unit demand between each pair of nodes) we have improved the bound to $O(1)$.

A related problem is the *minimum cost multicut*. Here we are given a weighted graph and a set of nodes. We need to find minimum cost set of edges whose deletion will disconnect all the nodes in the set. This type of problems are encountered in the context of network topology design and in the context of network reliability. A natural generalization is the *steiner multicut* problem. We are given a graph and several sets of nodes. We say that a set is "separated" if it is not contained in a single connected component. The problem is to find a minimum weight set of edges whose removal separates all of the given sets.

Using techniques used to produce the $O(\log k)$ approximation to the multicut problem [LR88, KRAR95, GVY93], it is relatively easy to get an $O(t \log k)$ approximation to the steiner multicut problem, where t is the cardinality of the largest set. In [KPRT97] we developed an $O(\log^3(kt))$ approximation algorithm.

The previous results about multicuts can be viewed as a consequence of a simple and very useful *network decomposition* lemma. This lemma states that that given a graph with n nodes and m edges, and a positive δ, one can remove at most $O(\frac{m}{\delta}\log n)$ edges such that the distance between any two nodes that stay in the same connected component is bounded by δ.[1] Other applications of network decomposition techniques of this type include routing with small size tables [PU89, ABLP90], symmetry-breaking [AGLP89], and synchronization of asynchronous networks [AP90].

In contrast to the previous approaches, our multicut results do not follow from a generalized decomposition lemma. We derive a generalization of the above lemma as a corollary of our steiner multicut theorem. We show that by removing at most $O(\frac{m}{\delta}\log^3(kt))$ edges, one can separate all the sets whose minimum steiner trees have at least δ edges, where k is the number of sets considered, t is the cardinality of the largest set, and m is the total number of edges. We also prove a weighted version of this lemma.

All of the above mentioned cut problems are for *undirected* graphs. The simplest directed problem is the *Feedback arc set*. This is the problem of finding a minimum set of edges whose removal makes the graph acyclic. Leighton and Rao showed an $O(\log^2 n)$ approximation algorithm [LR88], where n is the number of nodes. This was improved to $O(\log t \log\log t)$ by Seymour [Sey92], where t is the size of the minimum size feedback arc set. Seymour's result trivially extends to the weighted case and leads to an $O(\log n \log\log n)$ approximation algorithm (see [ENSS94]).

A natural generalization of the feedback arc set problem is to consider a weighted directed graph and a set of node pairs. The *directed multicut* problem is to find a minimum-weight multicut that separates all the given pairs. In other words, we have to find a set of edges whose removal ensures that none of the prespecified node pairs are contained in a strongly connected component.

An example of a problem that reduces to finding minimum weight directed multicuts is the *2-CNF clause-deletion* problem, i.e. the problem of finding a minimum weight set of clauses in a 2-CNF formula whose deletion makes the formula satisfiable. Previously known undirected multicut results have been used in approximation algorithm for the special case of 2-CNF$\equiv$ clause-deletion problem, as described in [KRAR95, GVY93].

As observed in [ENSS94], Seymour's result implies an $O(\log n \log\log n)$ approximation algorithm for the directed multicut problem.

In [KPRT97] we showed an $O(\log^2 k)$ approximation to the minimum weight directed multicut problem, where k is the number of given pairs. This is an improvement for the case where $k \ll n$.

At the heart of our minimum-weight directed multicut algorithm lies the following *directed decomposition* lemma: By removing at most $O(\frac{m}{\delta}\log^2 n)$ edges, any directed graph can be decomposed into a set of *strongly connected components* where any two nodes in the same connected component lie on a directed cycle whose length is at most δ.

The computational bottleneck of all the minimum cut approximation algorithms mentioned above is solving appropriately constructed linear programs. The LP needed to be solved for the directed cuts approximation can be solved by any linear programming algorithm. The corresponding LP in the steiner cuts case has an exponential number of constraints and can be solved by using convex programming (see *e.g.* [Vai89]). Although this leads to polynomial time algorithms, the resulting running time is pretty slow. Much faster algorithms can be obtained by using a general framework for solving packing and covering problems that we have developed in [PST91].

[1] Note that m/δ is an easy lower bound.

Publication during the reporting period

[AAF+97] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts. On-line machine scheduling with applications to load balancing and virtual circuit routing. *J. Assoc. Comput. Mach.*, 44(3):486–504, 1997.

[MP97] Y. Ma and S. Plotkin. Improved lower bounds for load balancing of tasks with unknown duration. *Information Processing Letters.* (62):301–303, 1997.

[KPRT97] P. Klein, S. Plotkin, S. Rao, and E. Tardos. Approximation algorithms for steiner and directed multicuts. *J. Alg.*, (22):241–269, 1997.

[Plot95] S. Plotkin. Competitive routing in ATM networks. *IEEE J. Selected Areas in Comm.*, pages 1128–1136, August 1995. Special issue on Advances in the Fundamentals of Networking. (Invited paper).

[PT95] S. Plotkin and É. Tardos. Improved bounds on the max-flow min-cut ratio for multicommodity flows. *Combinatorica.* 15(3):425–434, 1995.

[AKPPW95] Y. Azar, B. Kalyanasundaram, S. Plotkin, K. Pruhs, and O. Waarts. On-line load balancing of temporary tasks. *J. Alg.*, 1995.

[AAP93] Y. Afek, B. Awerbuch, S. Plotkin, and M. Saks. Local Management of a Global Resource in a Communication Network. *J. Assoc. Comput. Mach.*, 43(1):1–19, 1996.

[PST95] S. Plotkin, D. Shmoys, and É. Tardos. Fast approximation algorithms for fractional packing and covering problems. *Math of Oper. Research*, 20(2):257–301, 1995.

[GOPS98] A. Goldberg, J. D. Oldham, S. Plotkin, and C. Stein. An implementation of a combinatorial approximation algorithm for minimum-cost multicommodity flow. In *Proc. IPCO-98*, June 1998.

[GHP98] A. Goel, M. Henzinger, and S. Plotkin. Online throughput-competitive algorithm for multicast routing and admission control. In *Proc. 9th ACM-SIAM Symposium on Discrete Algorithms*, January 1998.

[KPP96] A. Kamath, O. Palmon, and S. Plotkin. Routing and admission control in general topology networks with poisson arrivals. In *Proc. 7th ACM-SIAM Symposium on Discrete Algorithms*, pages 269–278, 1996.

[KP95] D. Karger and S. Plotkin. Adding multiple cost constraints to combinatorial optimization problems, with applications to multicommodity flows. In *Proc. 27th Annual ACM Symposium on Theory of Computing*, pages 18–25, May 1995.

[KPP96] A. Kamath, O. Palmon, and S. Plotkin. Fast approximation algorithm for min-cost multicommodity flow. In *Proc. 6th ACM-SIAM Symposium on Discrete Algorithms*, 1995.

[GHP97] A. Goel, M. Henzinger, and S. Plotkin. Online throughput-competitive algorithm for multicast routing and admission control. Technical Report 97-1592, Stanford University, July 1997.

[KPP96] A. Kamath, O. Palmon, and S. Plotkin. Routing and admission control in general topology networks with poisson arrivals. Technical Report STAN-CS-TR-96-1575, Stanford University, 1996.

[MP96] Y. Ma and S. Plotkin. Improved lower bounds for load balancing of tasks with unknown duration. Technical Report STAN-CS-TN-96-37, Stanford University, 1996.

[GKPR95] R. Gawlick, A. Kamath, S. Plotkin, and K. Ramakrishnan. Routing and admission control in general topology networks. Technical Report STAN-CS-TR-95-1548, Stanford University, 1995.

[KPP95a] A. Kamath, O. Palmon, and S. Plotkin. Fast approximation algorithm for min-cost multicommodity flow. Technical Report STAN-CS-TN-95-19, Stanford University, 1995.

References

[AAF+97] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts. On-line machine scheduling with applications to load balancing and virtual circuit routing. *J. Assoc. Comput. Mach.*, 44(3):486–504, 1997.

[AAP93] B. Awerbuch, Y. Azar, and S. Plotkin. Throughput competitive on-line routing. In *Proc. 34th IEEE Annual Symposium on Foundations of Computer Science*, pages 32–40, November 1993.

[ABK92] Y. Azar, A. Broder, and A. Karlin. On-line load balancing. In *Proc. 33rd IEEE Annual Symposium on Foundations of Computer Science*, pages 218–225, 1992.

[ABLP90] Baruch Awerbuch, Amotz Bar-Noy, Nati Linial, and David Peleg. Improved routing strategies with succinct tables. *J. Algorithms*, 11:307–341, 1990.

[AGLP89] B. Awerbuch, A. Goldberg, M. Luby, and S. Plotkin. Network Decomposition and Locality in Distributed Computation. In *Proc. 30th IEEE Annual Symposium on Foundations of Computer Science*, pages 364–369, October 1989.

[AP90] Baruch Awerbuch and David Peleg. Network synchronization with polylogarithmic overhead. In *Proc. 31st IEEE Annual Symposium on Foundations of Computer Science*, pages 514–522, 1990.

[CN96] J. Castro and N. Nabona. An implementation of linear and nonlinear multicommodity network flows. *European Journal of Operational Research*, 92(1):37–53, July 1996.

[ENSS94] G. Even, J. Naor, B. Schieber, and M. Sudan. Approximating minimum feedback sets and multi-cuts in directed graphs. Unpublished manuscript, May 1994.

[GKPR94] R. Gawlick, A. Kamath, S. Plotkin, and K. Ramakrishnan. Distributed routing and admission control of virtual circuits in general topology networks. Technical Report BL011212-940819-19TM, AT&T Bell Laboratories, 1994.

[GOPS98] A. Goldberg, J. D. Oldham, S. Plotkin, and C. Stein. An implementation of a combinatorial approximation algorithm for minimum-cost multicommodity flow. In *Proc. IPCO-98*, June 1998.

[GVY93] N. Garg, V. V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. In *Proc. 25th Annual ACM Symposium on Theory of Computing*, May 1993.

[Kel86] F. P. Kelly. Blocking probabilities in large circuit-switched networks. *Adv. Appl. Prob.*, 18:473–505, 1986.

[KO88] K.R. Krishnan and T.J. Ott. Forward-looking routing: A new state-dependent routing scheme. In *Proc. of ITC-12, Torino, Italy*, 1988.

[KP95] D. Karger and S. Plotkin. Adding multiple cost constraints to combinatorial optimization problems, with applications to multicommodity flows. In *Proc. 27th Annual ACM Symposium on Theory of Computing*, pages 18–25, May 1995.

[KPP96] A. Kamath, O. Palmon, and S. Plotkin. Routing and admission control in general topology networks with poisson arrivals. In *Proc. 7th ACM-SIAM Symposium on Discrete Algorithms*, pages 269–278, 1996.

[KPRT97] P. Klein, S. Plotkin, S. Rao, and E. Tardos. Approximation algorithms for steiner and directed multicuts. *J. Alg.*, (22):241–269, 1997.

[KPST94] P. Klein, S. Plotkin, C. Stein, and É. Tardos. Faster approximation algorithms for the unit capacity concurrent flow problem with applications to routing and finding sparse cuts. *SIAM Journal on Computing*, 23(3):466–487, June 1994.

[KRAR95] P. N. Klein, S. Rao, A. Agrawal, and R. Ravi. An approximate max-flow min-cut relation for multicommodity flow, with applications. *Combinatorica*, 15(2):187–202, 1995. Preliminary version appeared as "Approximation through multicommodity flow," In *Proc. 31th IEEE Annual Symposium on Foundations of Computer Science*, pages 726–737, 1990.

[KV86] S. Kapoor and P. M. Vaidya. Fast Algorithms for Convex Quadratic Programming and Multicommodity Flows. In *Proc. 18th Annual ACM Symposium on Theory of Computing*, pages 147–159, 1986.

[LMP+95] T. Leighton, F. Makedon, S. Plotkin, C. Stein, É. Tardos, and S. Tragoudas. Fast approximation algorithms for multicommodity flow problems. *J. Comp. and Syst. Sci.*, 50:228–243, 1995. (Invited paper).

[LR88] T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *Proc. 29th IEEE Annual Symposium on Foundations of Computer Science*, pages 422–431, 1988.

[MP97] Y. Ma and S. Plotkin. Improved lower bounds for load balancing of tasks with unknown duration. *Information Processing Letters*, (62):301–303, 1997.

[OK85] T.J. Ott and K.R. Krishnan. State-dependent routing of telephone traffic and the use of separable routing schemes. In *Proc. of ITC-11, Kyoto, Japan*, 1985.

[PST91] S. Plotkin, D. Shmoys, and É. Tardos. Fast approximation algorithms for fractional packing and covering problems. In *Proc. 32nd IEEE Annual Symposium on Foundations of Computer Science*, pages 495–504, October 1991.

[PST95] S. Plotkin, D. Shmoys, and É. Tardos. Fast approximation algorithms for fractional packing and covering problems. *Math of Oper. Research*, 20(2):257–301, 1995.

[PT95] S. Plotkin and É. Tardos. Improved bounds on the max-flow min-cut ratio for multicommodity flows. *Combinatorica*, 15(3):425–434, 1995.

[PU89] D. Peleg and E. Upfal. A tradeoff between size and efficiency for routing tables. *Journal of the ACM*, 36:510–530, 1989.

[Sey92] P.D. Seymour. Packing directed circuits fractionally. Unpublished manuscript. Revised November 1993, June 1992.

[Vai87] P. M. Vaidya. An algorithm for linear programming that requires $O(((m+n)n^2+(m+n)^{1.5}n)L)$ arithmetic operations. In *Proc. 19th Annual ACM Symposium on Theory of Computing*, pages 29–38, 1987.

[Vai89] P.M. Vaidya. A new algorithm for minimizing convex functions over convex sets. In *Proc. 30th IEEE Annual Symposium on Foundations of Computer Science*, pages 338–343, 1989.